

IN THE CLAIMS:

Claims 1-12 canceled.

1 13. (Previously presented) A method for programmably allocating resources to
2 accommodate I/O transactions at I/O ports of a multiprocessor computer system compris-
3 ing:
4 determining the number of devices being serviced via the ports,
5 identifying at least one assembly for hot swapping,
6 copying the contents of cache memories associated with the at least one identified
7 assembly,
8 setting criteria for transactions at the port with respect to the number of devices,
9 and
10 with respect to the numbers of devices at the ports, assigning resources to the
11 ports.

1 14. (Previously presented) The method as defined in claim 13 wherein assigning
2 resources to the ports comprises at least one of assigning control registers to the ports,
3 assigning direct memory access engines to the ports, assigning cache memory to the ports
4 and assigning priorities among the transactions at the ports.

1 15. (Previously presented) A system for programmably allocating resources to ac-
2 commodate I/O transactions at I/O ports of a multiprocessor computer system, the system
3 comprising:
4 means for determining the number of devices being serviced via a port,

5 at least one assembly identified for hot swapping,
6 means for copying the contents of cache memories associated with the at least one
7 identified assembly,
8 means for setting criteria for transactions at the port with respect to the number of
9 devices, and
10 means, responsive to the criteria, for assigning resources to the ports.

1 16. (Previously presented) The system as defined in claim 15 wherein the re-
2 sources assigned to the ports comprises at least one of
3 direct memory access (DMA) engines,
4 cache memory, and
5 means for assigning priorities among the transactions at the ports.

1 17 (Previously presented) The method as defined in claim 13 further comprising
2 determining the number and types of transactions anticipated at the ports, wherein the
3 assignment of resources is further with respect to the numbers and types of transactions at
4 the ports.

1 18. (Previously presented) The method as defined in claim 13 wherein the at least
2 one identified assembly has a memory system, and the method further comprises copying
3 the states and status of the memory systems associated with at least one identified assem-
4 bly.

1 19. (Previously presented) The system as defined in claim 15 further comprising
2 means for determining the number and types of transactions anticipated at the ports,
3 wherein the criteria further accounts for the anticipated number and types of transactions.

1 20. (Previously presented) The system as defined in claim 15 wherein the at least
2 one identified assembly has a memory system, and the system further comprises means
3 for copying the states and status of the memory systems associated with the at least one
4 identified assembly.

1 21. (Previously presented) A method for programmably allocating resources for
2 processing Input/Output (I/O) transactions at a plurality of I/O ports of an I/O bridge, the
3 method comprising:

4 identifying the number of I/O devices being serviced by at least one I/O port;
5 setting criteria for the transactions at the at least one I/O port with respect to the
6 number of I/O devices being serviced by the port; and
7 assigning the resources to the at least one I/O port in response to the criteria.

1 22. (Previously presented) The method of claim 21 wherein the assigning com-
2 prises assigning a plurality of direct memory access (DMA) engines for use in processing
3 I/O transactions.

1 23. (Previously presented) The method of claim 22 wherein assigning comprises
2 apportioning a selected number of DMA engines to process a given transaction at a par-
3 ticular I/O port.

1 24. (Previously presented) The method of claim 22 wherein assigning comprises
2 apportioning at least one DMA engine to process at least one transaction at a port.

1 25. (Previously presented) The method of claim 22 wherein assigning comprises
2 apportioning one DMA engine to process a given transaction at a port identified as ser-
3 vicing multiple I/O devices.

1 26. (Previously presented) The method of claim 21 wherein assigning comprises
2 assigning at least one miss address file (MAF) value for processing I/O transactions.

1 27. (Previously presented) The method of claim 21 wherein assigning comprises
2 assigning a plurality of miss address file (MAF) values for processing I/O transactions.

1 28. (Previously presented) The method of claim 27 further comprising reducing
2 the assigned number of MAF values.

1 29. (Previously presented) The method of claim 21 wherein
2 the I/O bridge is configured to utilize a plurality of virtual channels to communi-
3 cate with at least one processors of a multiprocessor computer system, and
4 the resources include flow control credits associated with each of the plurality of
5 virtual channels.

1 30. (Previously presented) The method of claim 29 wherein assigning comprises
2 setting the number of flow control credits associated with each virtual channel.

1 31. (Previously presented) The method of claim 21 wherein
2 the I/O bridge comprises at least one control register, the at least one control reg-
3 ister having a plurality of fields, and at least one field of the control register being associ-
4 ated with a corresponding resource, and
5 the method further comprises writing to a selected field of the at least one control
6 register so as to modify the assignment of resources.

1 32. (Previously presented) An Input/Output (I/O) bridge for use in a computer
2 system having a plurality of processors, the I/O bridge comprising:

3 a plurality of I/O ports, each I/O port configured to communicate with at least one
4 I/O device that generates or receives transactions;
5 resources for use in servicing the transactions of the I/O devices; and
6 programmable logic configured and arranged to assign the resources among the
7 I/O ports in response to the number of I/O devices with which the I/O ports are commu-
8 nicating.

1 33. (Previously presented) The I/O bridge of claim 32 wherein
2 the resources comprise at least one direct memory access (DMA) engine config-
3 ured to process the transactions, and
4 the programmable logic apportions the at least one of DMA engine to process at
5 least one transaction at a given I/O port in response to the number of I/O devices coupled
6 to the given I/O port.

1 34. (Previously presented) The I/O bridge of claim 32 wherein
2 the resources include a plurality of miss address file (MAF) values for use in re-
3 questing information from the computer system, and
4 the programmable logic sets the number of available MAF values.

1 35. (Previously presented) The I/O bridge of claim 32 wherein
2 the I/O bridge communicates with the computer system through a plurality of vir-
3 tual channels,
4 the resources include a plurality of flow control credits associated with the virtual
5 channels, and

6 the programmable logic assigns a number of flow control credits to each virtual
7 channel.

1 36. (Previously presented) the I/O bridge of claim 35 wherein the virtual channels
2 comprise a Request channel, a Read I/O channel, and a Write I/O channel.

1 37. (Previously presented) The I/O bridge of claim 33 further comprising at least
2 one cache for storing information, wherein, to hot-swap an assembly of the computer sys-
3 tem, the programmable logic is configured to

4 disable the at least one DMA engine, and
5 flush the information from the at least one cache.

1 38. (Previously presented) The I/O bridge of claim 37 wherein the at least one
2 cache is one of a write cache, a read cache and a translation look-aside buffer (TLB).

1 39. (Previously presented) The I/O bridge of claim 37 wherein the assembly is a
2 processor.

1 40. (Previously presented) The I/O bridge of claim 33 wherein
2 the programmable logic comprises at least one control register associated with
3 each I/O port, and

4 the at least one control register has a first field for apportioning the at least one
5 DMA engine.

1 41. (Previously presented) The I/O bridge of claim 32 wherein the programmable
2 logic re-assigns resources among the I/O ports dynamically while the I/O bridge contin-
3 ues to operate.